In [1]:
```
#導入需要的依賴包
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

#統計預測圖

seaborn.relplot() 是一個非常好用的繪圖指令, 可以一次把數個變數的關係呈現在一張圖表上.

`load_dataset` looks for online csv files on https://github.com/mwaskom/seaborn-data (https://github.com/mwaskom/seaborn-data). Here's the docstring:

Load a dataset from the online repository (requires internet).

> Parameters

> name : str Name of the dataset (name.csv on https://github.com/mwaskom/seaborn-data (https://github.com/mwaskom/seaborn-data)). You can obtain list of available datasets using func:get_dataset_names
> kws : dict, optional Passed to pandas.read_csv

In [90]:
```
#加載sns內置數據集
tips=sns.load_dataset('tips')
sns.set(color_codes=True)
tips.head(3)
```

Out[90]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3

內建數據集-tips數據集

> 關於數據集，total_bill是消費總金額，tip是小費，size指用餐人數。

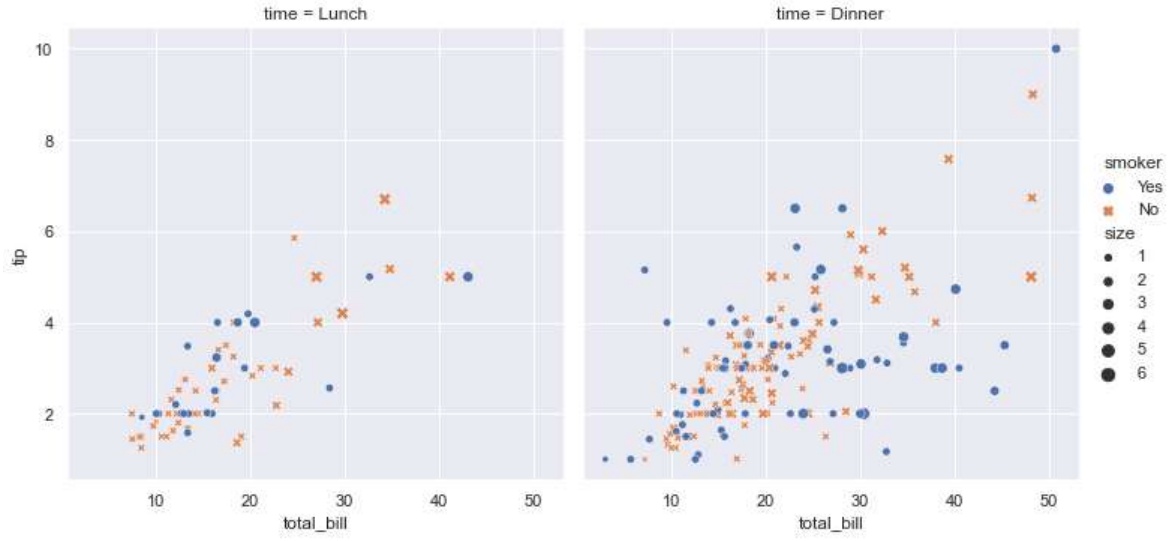
In [95]:
```
#NBA = sns.load_dataset('data/NBA_season1718_salary.csv')

# Path of the file to read
fifa_filepath = 'data/NBA_season1718_salary.csv'

# Readd the file into a variable fifa_data
fifa_data = pd.read_csv(fifa_filepath, index_col="Unnamed: 0", parse_dates

# Print the first 5 rows of the data
fifa_data.head()
```

Out[95]:

	Player	Tm	season17_18
1	Stephen Curry	GSW	34682550.0
2	LeBron James	CLE	33285709.0
3	Paul Millsap	DEN	31269231.0
4	Gordon Hayward	BOS	29727900.0
5	Blake Griffin	DET	29512900.0

In [4]:
```
sns.relplot(x='total_bill',y='tip',col='time',hue='smoker',
    style='smoker',size='size',data=tips
)
```

Out[4]: <seaborn.axisgrid.FacetGrid at 0x1db31b02d30>



#單變量分析繪圖（直方圖、條形圖）

Dataset: NBA_season1718_salary.csv
ref:https://www.kaggle.com/koki25ando/salary (https://www.kaggle.com/koki25ando/salary)

In [7]:
```
data=pd.read_csv('data/NBA_season1718_salary.csv')
data
```

Out[7]:

	Unnamed: 0	Player	Tm	season17_18
0	1	Stephen Curry	GSW	34682550.0
1	2	LeBron James	CLE	33285709.0
2	3	Paul Millsap	DEN	31269231.0
3	4	Gordon Hayward	BOS	29727900.0
4	5	Blake Griffin	DET	29512900.0
...	...	...	...	...
568	569	Quinn Cook	NOP	25000.0
569	570	Chris Johnson	HOU	25000.0
570	571	Beno Udrih	DET	25000.0
571	572	Joel Bolomboy	MIL	22248.0
572	573	Jarell Eddie	CHI	17224.0

573 rows × 4 columns

In [10]:
```
#對讀進來的數據按薪資降序排序並取前10條數據
salary_top10=(data.sort_values('season17_18',ascending=False)).head(10)
salary_top10
```

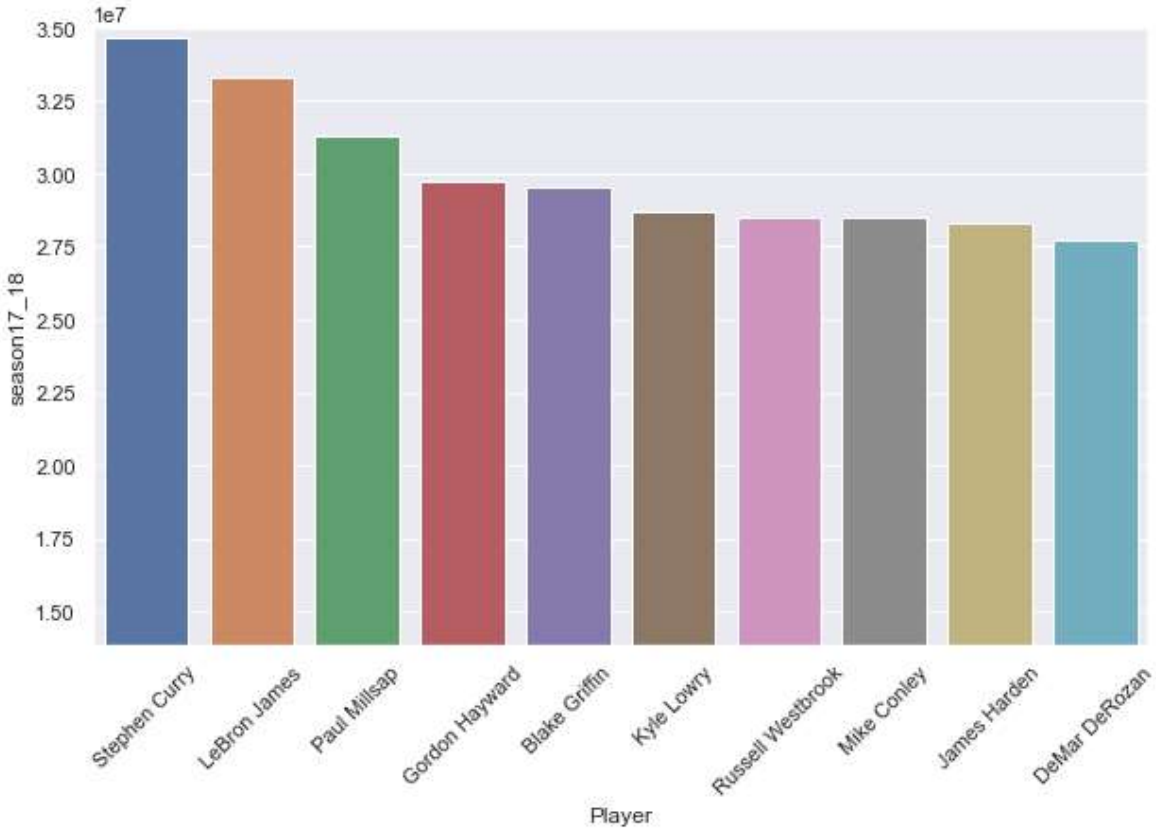
Out[10]:

	Unnamed: 0	Player	Tm	season17_18
0	1	Stephen Curry	GSW	34682550.0
1	2	LeBron James	CLE	33285709.0
2	3	Paul Millsap	DEN	31269231.0
3	4	Gordon Hayward	BOS	29727900.0
4	5	Blake Griffin	DET	29512900.0
5	6	Kyle Lowry	TOR	28703704.0
6	7	Russell Westbrook	OKC	28530608.0
7	8	Mike Conley	MEM	28530608.0
8	9	James Harden	HOU	28299399.0
9	10	DeMar DeRozan	TOR	27739975.0

In [11]:
```
#取出球員列數據，作為條形圖橫坐標
player=salary_top10['Player']


#取出球員薪資列，作為條形圖縱坐標
season_salary=salary_top10['season17_18']
```

In [16]:
```
plt.figure(figsize=(10,6))
plt.xticks(rotation=45)
plt.ylim(season_salary.min()*0.5,season_salary.max()*1.01)

#ci參數表示允許的誤差範圍（控制誤差棒的百分比，在0-100之間）
sns.barplot(x=player,y=season_salary,data=salary_top10,ci=68)
```

Out[16]: <AxesSubplot:xlabel='Player', ylabel='season17_18'>



Result

從條形圖中不難看出，勇士隊的當家球星curry資最高，將近達到3500萬美元，位居第二的是勒布朗詹姆斯，也是在3300萬美元以上。上面的條形圖是按球員作為橫坐標，下面我們按球隊匯總，看看哪個球隊的薪資總額最高。

練習 - 彙總球隊薪資繪製成直方圖

按球隊匯總，看看哪個球隊的薪資總額最高。

In [22]:
```
salary_by_team=data.groupby(by='Tm').sum()
salary_by_team
```

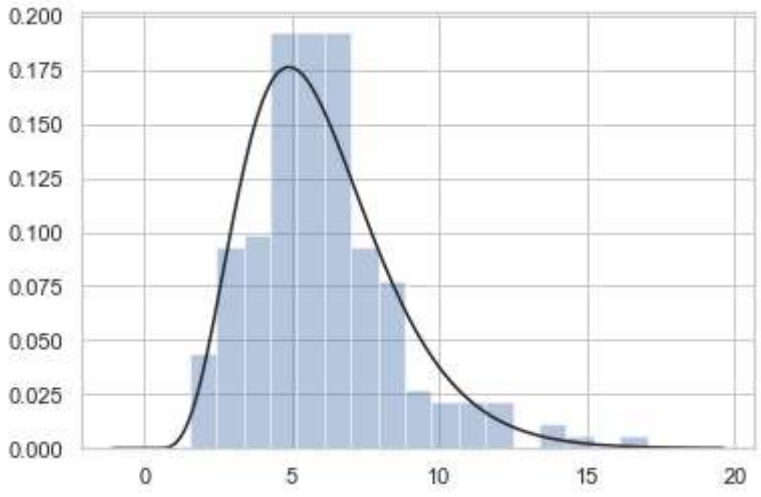
Out[22]:

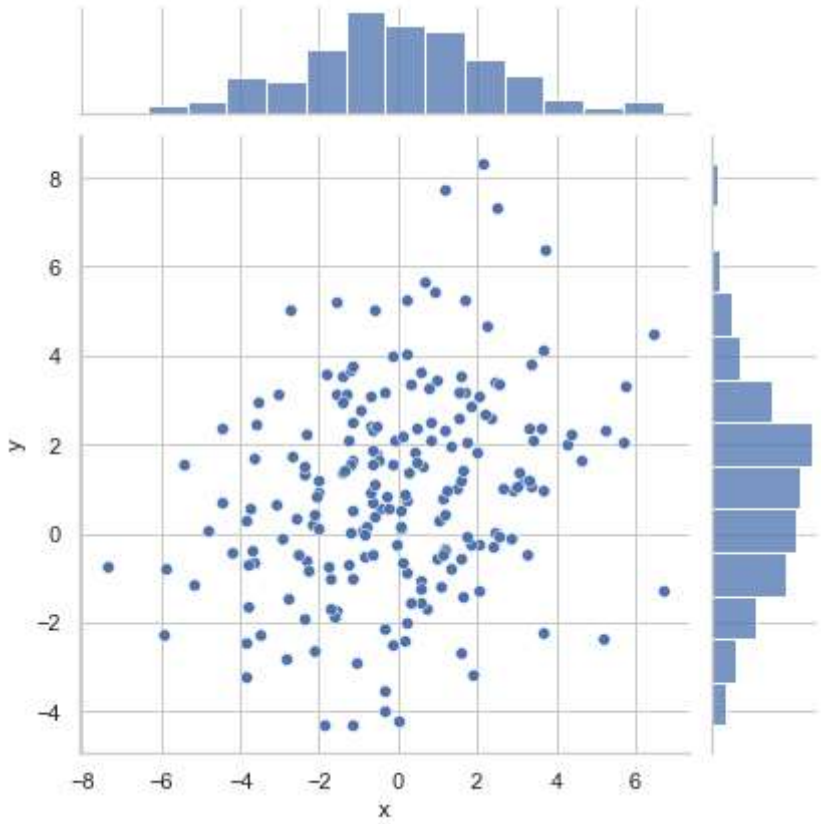
Tm	Unnamed: 0	season17_18
ATL	9324	100217797.0
BOS	4712	115009962.0
BRK	4953	96039772.0
CHI	7551	89425042.0
CHO	3061	117228164.0
CLE	2979	137288549.0
DAL	7636	85821361.0
DEN	3624	107889099.0
DET	5639	120086105.0
GSW	3892	137494845.0
HOU	7363	119024081.0
IND	5653	94429791.0
LAC	5891	118907163.0
LAL	7241	105355450.0
MEM	6557	110104061.0
MIA	4392	131222624.0
MIL	6485	120686318.0
MIN	3482	117468554.0
NOP	6775	119725571.0
NYK	6024	107855405.0
OKC	3246	134294056.0
ORL	4567	96925954.0
PHI	6194	100794278.0
PHO	6290	94818333.0
POR	5307	119108924.0
SAC	5164	95627528.0
SAS	4467	116153554.0
TOR	5109	116528796.0
UTA	5855	107465014.0
WAS	5018	124179842.0

In [29]:
```
#先將資料集用groupby根據球隊分別匯總
salary_by_team=data.groupby(by='Tm').sum().reset_index()
salary_by_team=salary_by_team.sort_values('season17_18',ascending=False)
tm_top10=salary_by_team[1:10]
tm=tm_top10['Tm']
salary_sum=tm_top10['season17_18']

#x=tm ,y=salary_sum
sns.barplot(tm,salary_sum,data=tm_top10)
sns.color_palette('hls',10)
sns.set(style='whitegrid')
```

In [34]:
```
import scipy.stats as stats
#畫圖1
x=np.random.gamma(6,size=200)
sns.distplot(x,kde=False,fit=stats.gamma)
#畫圖2
mean,cov=[0,1],[(1,5),(5,1)]
data=np.random.multivariate_normal(mean,cov,200)
df=pd.DataFrame(data,columns=['x','y'])
sns.jointplot(x='x',y='y',data=df)
```

Out[34]: <seaborn.axisgrid.JointGrid at 0x1db39db0550>





#Seaborn回歸分析繪圖(regplot)

In [88]:
```
#導入依賴包
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
#加載sns內置數據集
tips=sns.load_dataset('tips')
sns.set(color_codes=True)
tips.head(3)
```

Out[88]:

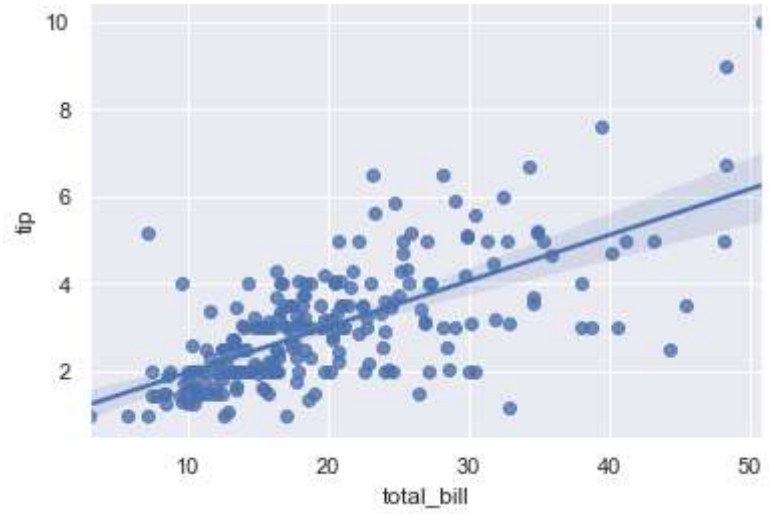
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3

內建數據集-tips數據集

> 關於數據集，total_bill是消費總金額，tip是小費，size指用餐人數。

In [15]:
```
sns.regplot(data=tips,x='total_bill',y='tip')
```

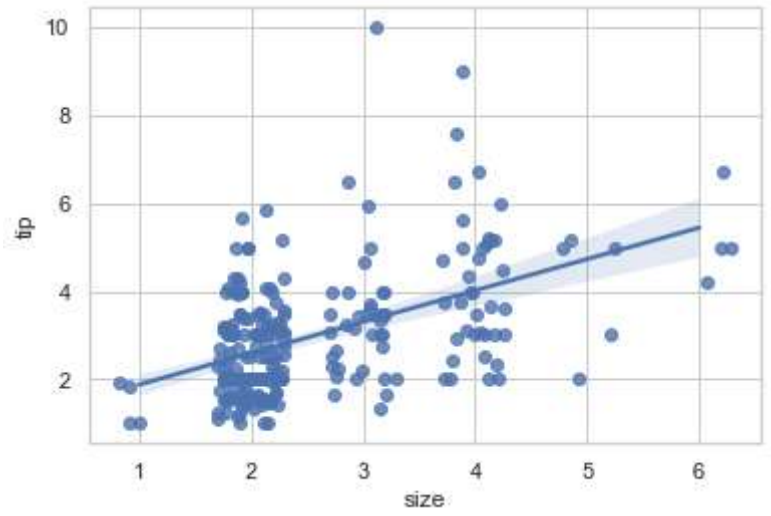
Out[15]: <AxesSubplot:xlabel='total_bill', ylabel='tip'>



[練習]看看其他變量和小費之間的關係。

In [37]:
```
sns.regplot(data=tips,x='size',y='tip',x_jitter=0.3)
```

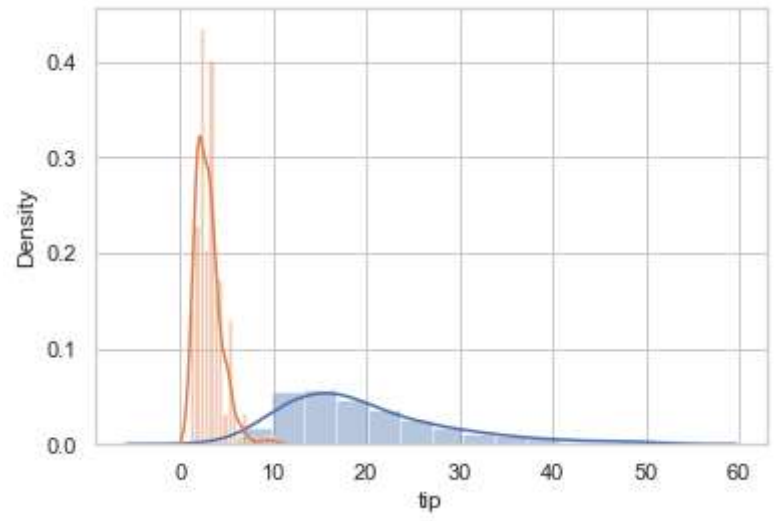
Out[37]: <AxesSubplot:xlabel='size', ylabel='tip'>



In [38]:
```
#導入依賴包 %matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="whitegrid", color_codes=True)
tips = sns.load_dataset("tips")
```

In [39]:
```
sns.distplot(tips['total_bill'])
sns.distplot(tips['tip'])
```

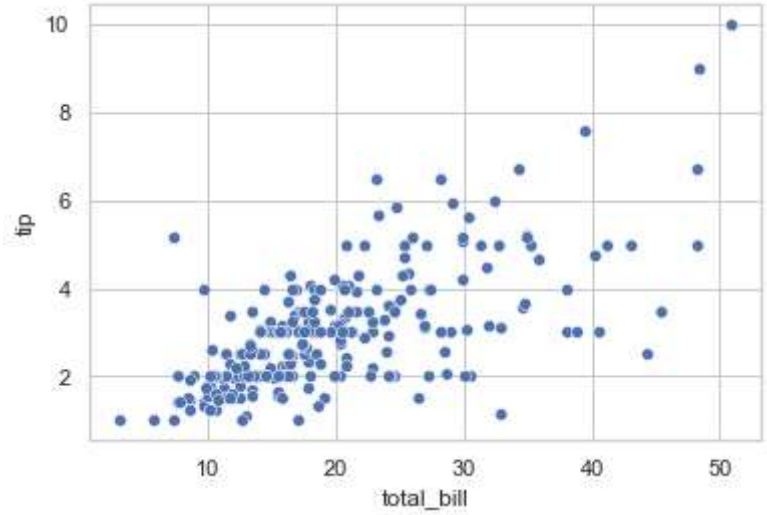
Out[39]: <AxesSubplot:xlabel='tip', ylabel='Density'>



#散點圖（分布散點、分簇散點圖）

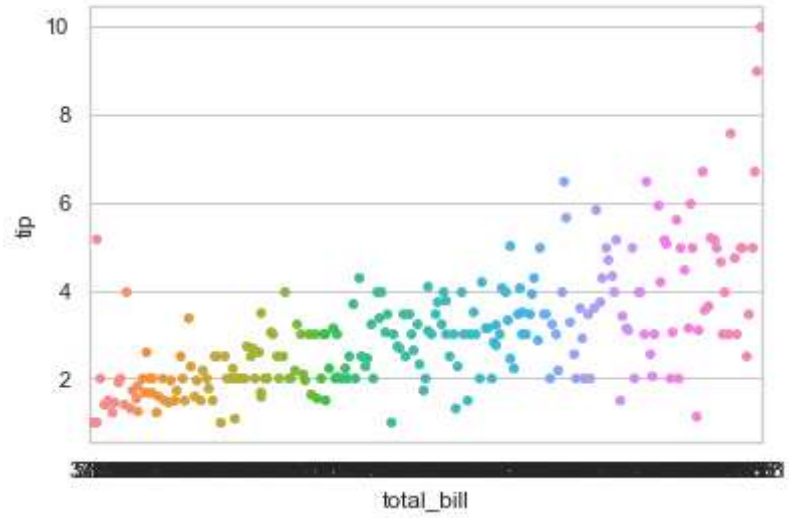
In [40]:
```
sns.scatterplot(data=tips,x='total_bill',y='tip')
```

Out[40]: <AxesSubplot:xlabel='total_bill', ylabel='tip'>



In [44]:
```
sns.stripplot(x='total_bill',y='tip',data=tips)
```

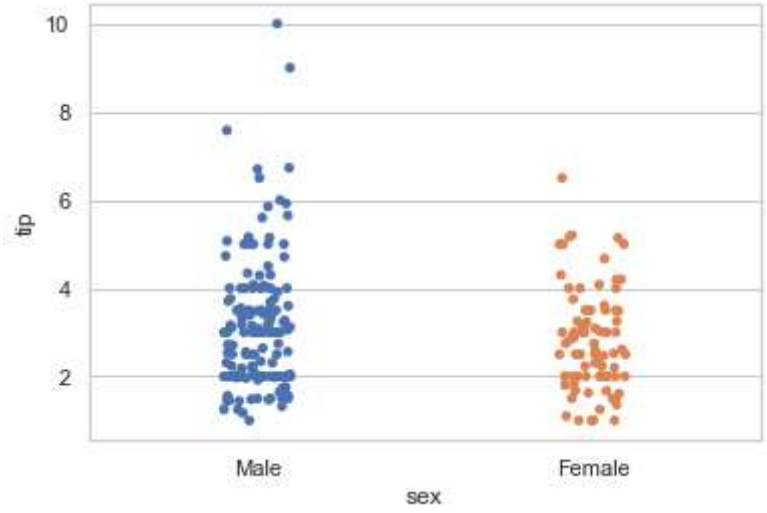
Out[44]: <AxesSubplot:xlabel='total_bill', ylabel='tip'>



從上面的圖中可以看到，stripplot參數中的x是橫軸數據（分類數據），y是縱軸數據（類型對應數據），由於我們將總金額作為分類，導致分類太多，橫軸數據根本看不清，所以我們重新傳參，分別看下性別和小費、時間和小費，用餐日期跟小費之間的數據分布。

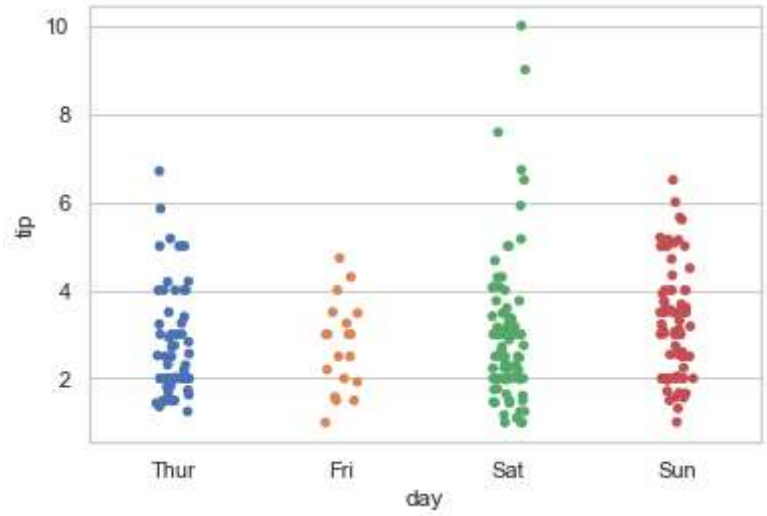
In [45]:
```
sns.stripplot(x='sex',y='tip',data=tips)
```

Out[45]: <AxesSubplot:xlabel='sex', ylabel='tip'>



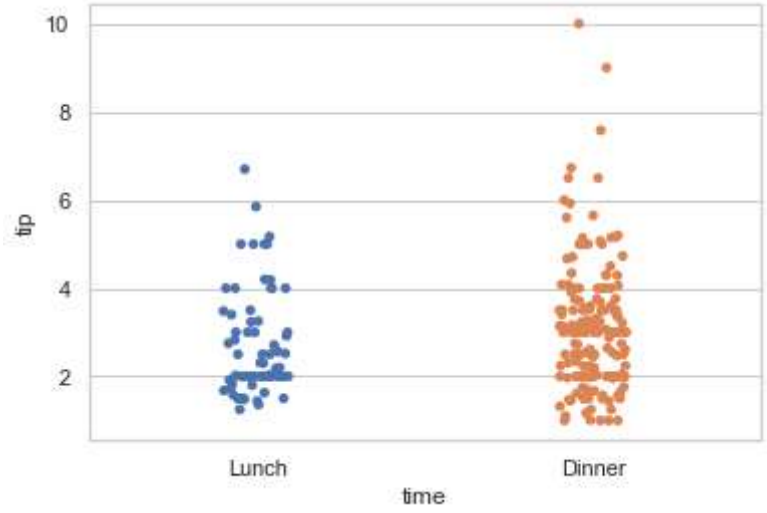
In [46]:
```
sns.stripplot(x='day',y='tip',data=tips)
```

Out[46]: <AxesSubplot:xlabel='day', ylabel='tip'>



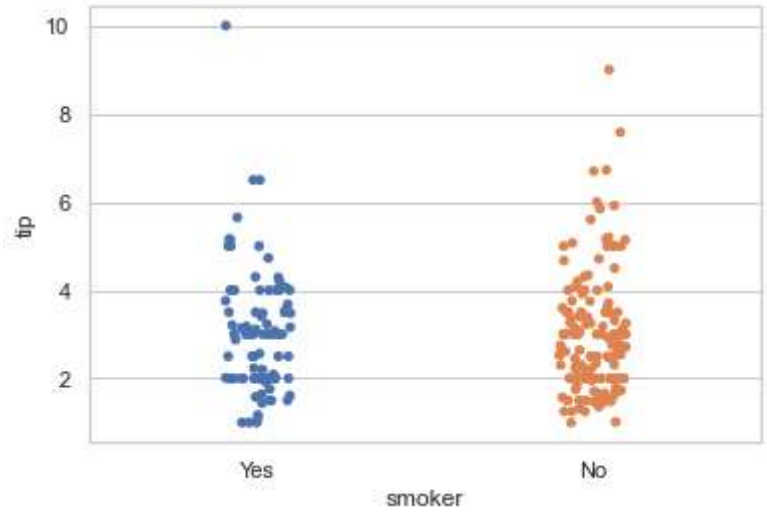
In [47]:
```
sns.stripplot(x='time',y='tip',data=tips)
```

Out[47]:
```
<AxesSubplot:xlabel='time', ylabel='tip'>
```



In [48]:
```
sns.stripplot(x='smoker',y='tip',data=tips)
```

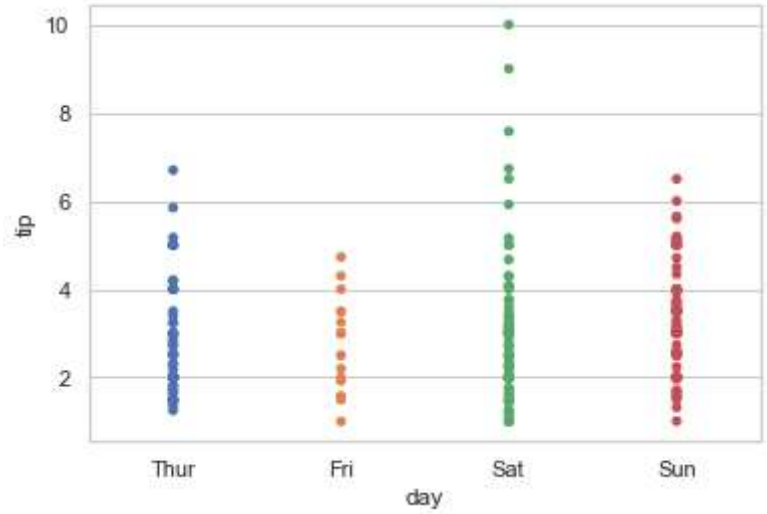
Out[48]:
```
<AxesSubplot:xlabel='smoker', ylabel='tip'>
```



因為jitter（數據抖動）這個參數的默認值是True,如果我們將其設置為False，效果如圖：

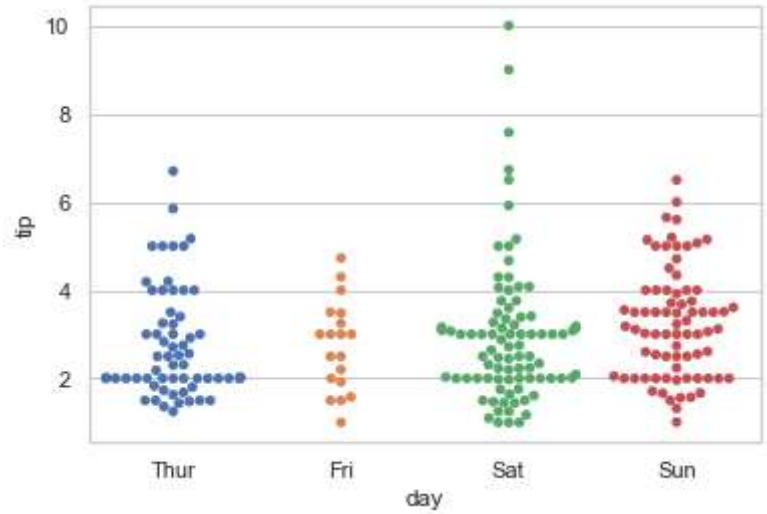
In [50]:
```
#jitter(數據抖動) = False
sns.stripplot(x='day',y='tip',jitter=False,data=tips)
```

Out[50]: <AxesSubplot:xlabel='day', ylabel='tip'>



In [52]:
```
sns.swarmplot(x='day',y='tip',data=tips)
```

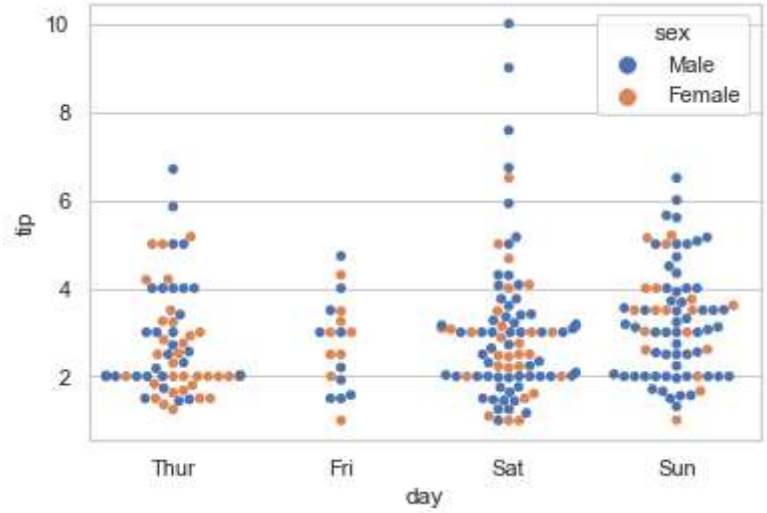
Out[52]: <AxesSubplot:xlabel='day', ylabel='tip'>



在stripplot函數中，我們只看了某一類別對應的數據，而沒有看類中對應其他類的數據分布，舉個例子，比如我們用stripplot看到男性和女性對應的小費數據，但是不知道男性和女性對應的數據中哪些是吸菸的，哪些是不吸菸的，對於這個問題這兩個函數都有一個hue參數，這裡我們就以分簇散點圖函數為例，演示下這個參數的效果，如圖：

In [53]:
```
sns.swarmplot(x='day',y='tip',hue='sex',data=tips)
```

Out[53]: <AxesSubplot:xlabel='day', ylabel='tip'>



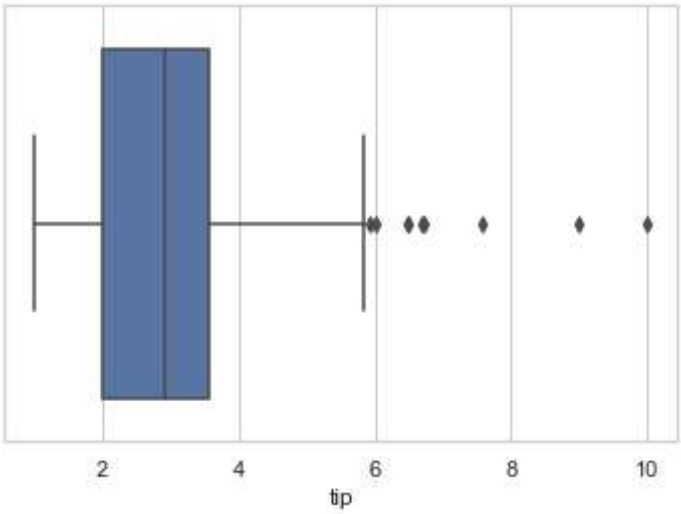
上面這個圖就很好的展示了不同性別在不同的星期中付小費數據的分布。

##繪製盒圖、小提琴圖

seaborn.boxplot()
這個函數主要是繪製出一個箱型圖來反映離群點數據。

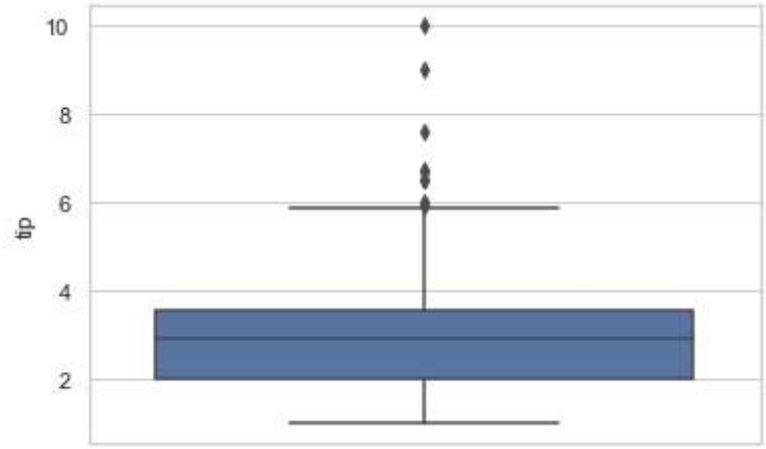
In [54]:
```
sns.boxplot(x='tip',data=tips)
```

Out[54]: <AxesSubplot:xlabel='tip'>



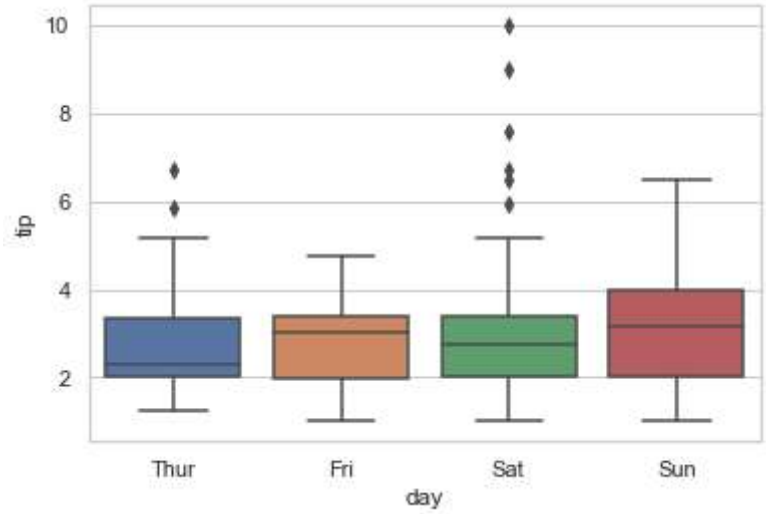
In [34]:
```
sns.boxplot(y='tip',data=tips)
```

Out[34]: <AxesSubplot:ylabel='tip'>



In [55]:
```
sns.boxplot(x='day',y='tip',data=tips)
```

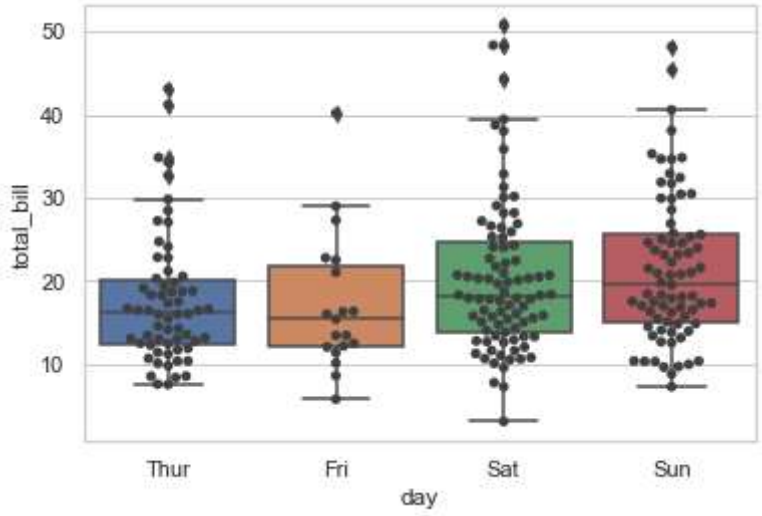
Out[55]: <AxesSubplot:xlabel='day', ylabel='tip'>



seaborn庫中的很多繪圖方法就是提前給你畫好模板，我們只需要把數據傳進去就得到相應的圖形， 基於此，我門是不是可以在同一個模板上同時繪製分簇圖和箱型圖呢？

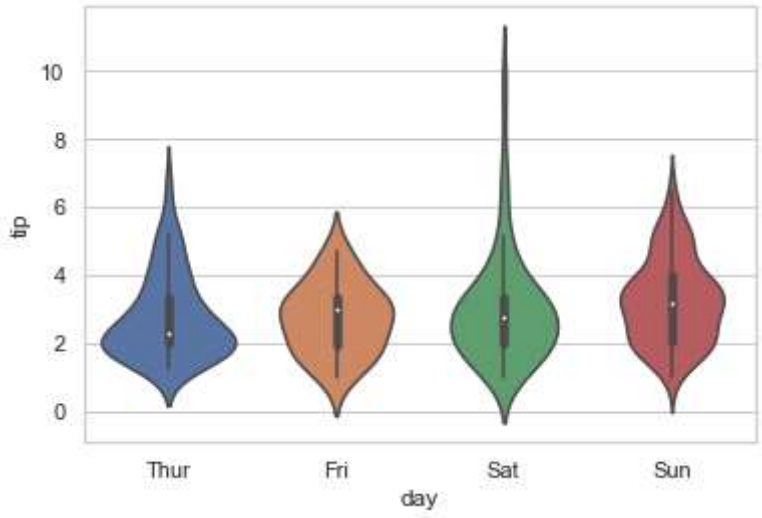
In [57]:
```
#圖重疊
sns.boxplot(x='day',y='total_bill',data=tips)
sns.swarmplot(x='day',y='total_bill',data=tips,color='.25')
```

Out[57]: <AxesSubplot:xlabel='day', ylabel='total_bill'>



In [60]:
```
sns.violinplot(x='day',y='tip',data=tips)
```

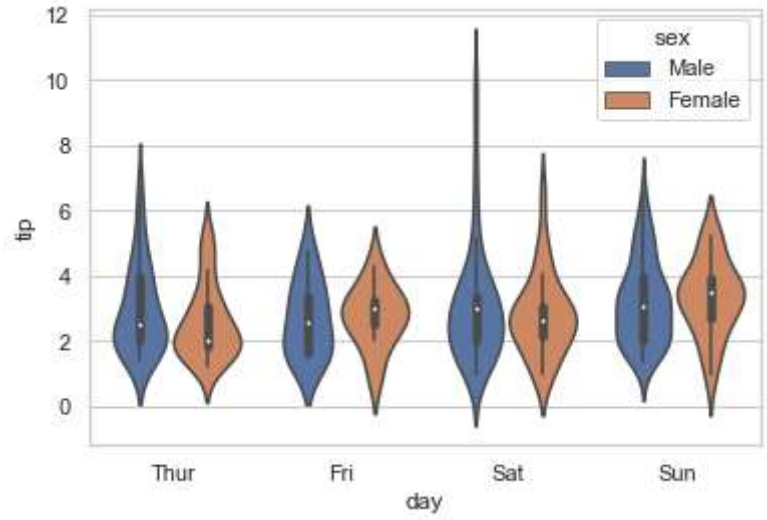
Out[60]: <AxesSubplot:xlabel='day', ylabel='tip'>



小提琴圖即反映了數據的離群情況，同時也反映了數據的分布密度。
此函數跟箱型圖一樣，也有hue參數，可以查看第二個分類屬性下的數據分布，如下：

In [62]:
```
#多 Hue
sns.violinplot(x='day',y='tip',data=tips,hue='sex')
```

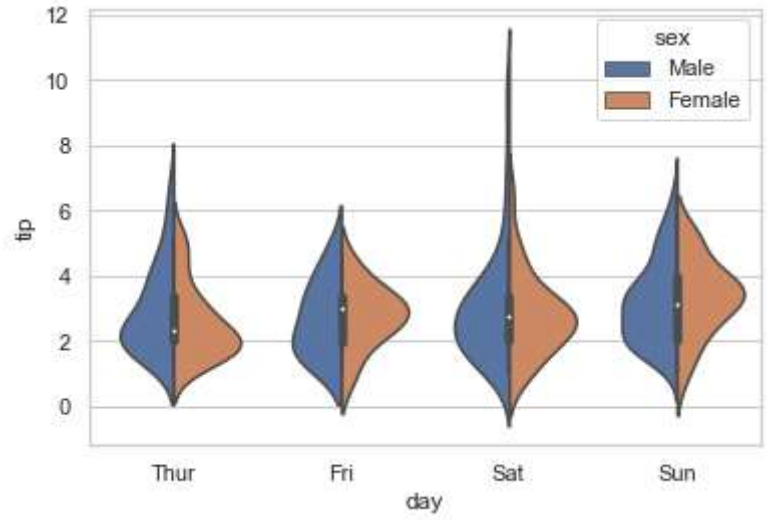
Out[62]: <AxesSubplot:xlabel='day', ylabel='tip'>



可以看到，上面的小提琴圖是4對8個，即當設置hue參數為sex時，每個小提琴的數據集是一類數據，那有沒有辦法讓男性對應數據和女性對應數據都作為一個數據集繪製出一個小提琴圖，只是在一個圖中區分男性數據和女性數據呢？答案是肯定的，violinplot()函數提供了一個split參數，可以做到以上需求，如下：

In [64]:
```
sns.violinplot(x='day',y='tip',data=tips,hue='sex',split=True)
```

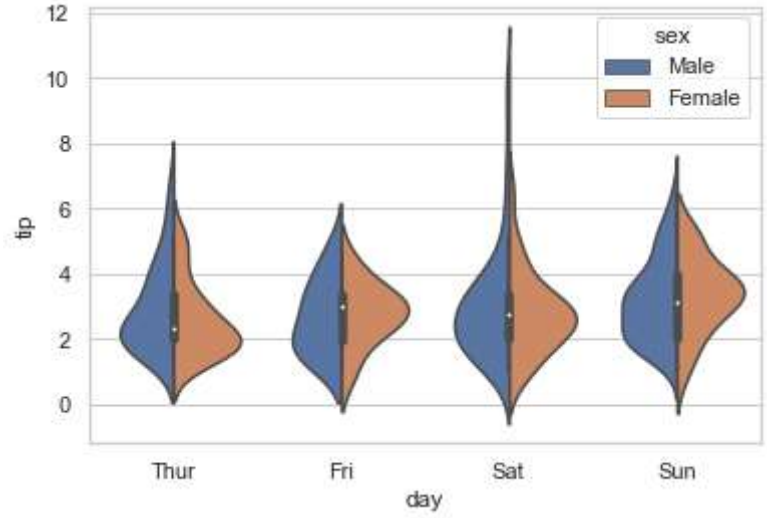
Out[64]: <AxesSubplot:xlabel='day', ylabel='tip'>



除此以外，我們再來看一下inner參數，上面的小提琴圖中內部都是箱型圖，通過inner參數，我們可以設置其他類型，inner可選參數有：box、quartile(四分位)、point、stick。我們依次看下：

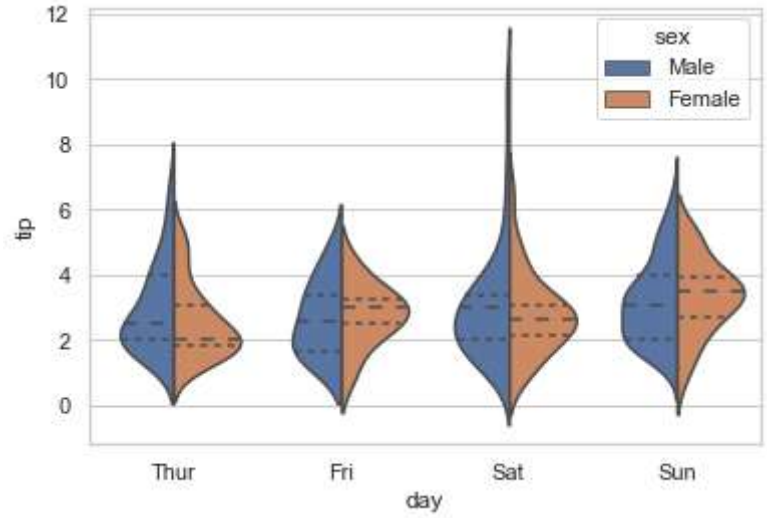
In [65]:
```
ns.violinplot(x='day',y='tip',data=tips,hue='sex',split=True,inner='box')
```

Out[65]: <AxesSubplot:xlabel='day', ylabel='tip'>



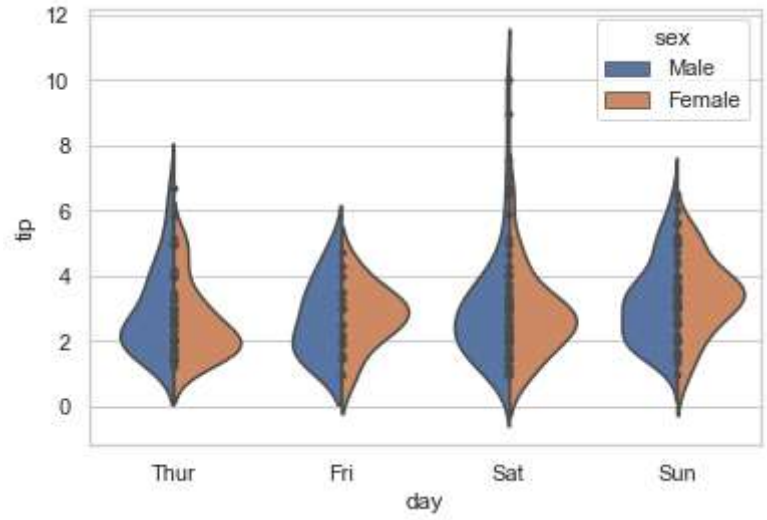
In [66]:
```
olinplot(x='day',y='tip',data=tips,hue='sex',split=True,inner='quartile')
```

Out[66]: <AxesSubplot:xlabel='day', ylabel='tip'>



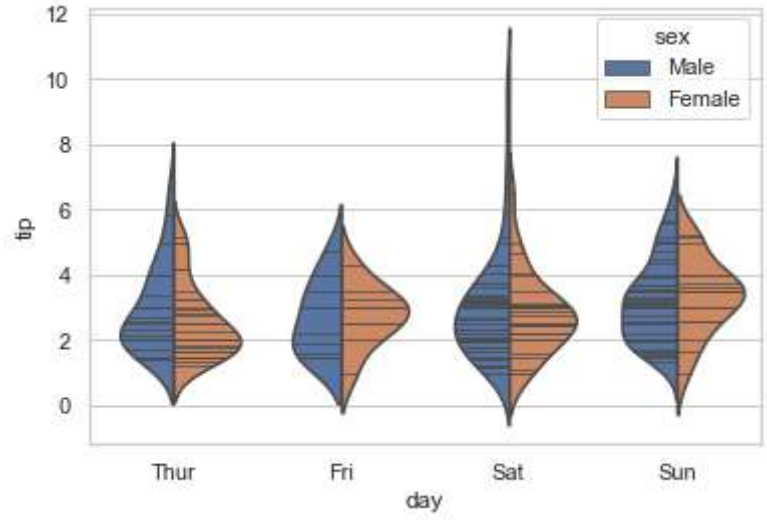
In [67]:
```
sns.violinplot(x='day',y='tip',data=tips,hue='sex',split=True,inner='point
```

Out[67]: <AxesSubplot:xlabel='day', ylabel='tip'>



In [68]:
```
.violinplot(x='day',y='tip',data=tips,hue='sex',split=True,inner='stick')
```

Out[68]: <AxesSubplot:xlabel='day', ylabel='tip'>

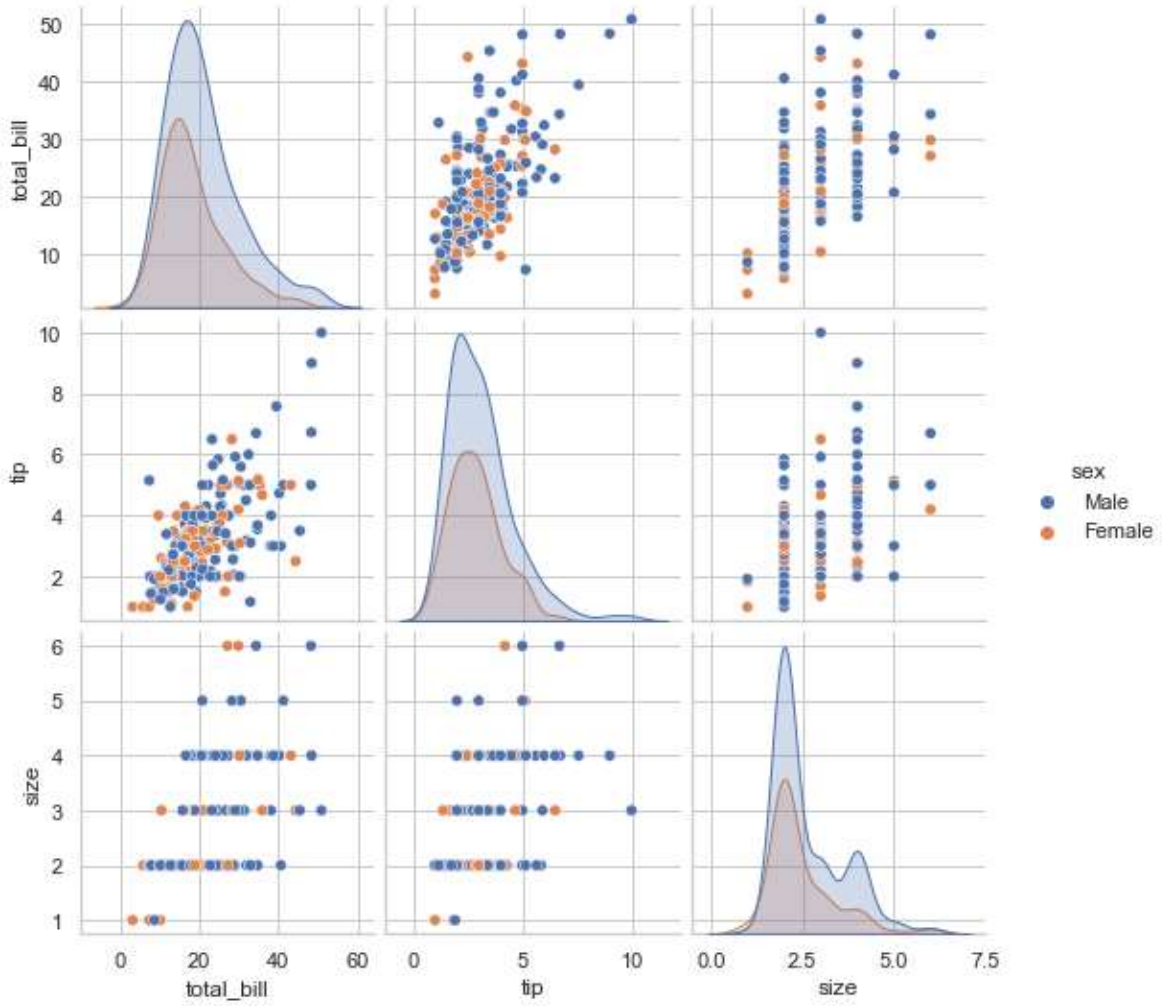


#結構化展示多維數據

Visualizing the multidimensional relationships among the samples is as easy as calling sns.pairplot:

In [69]:
```
sns.pairplot(tips,hue='sex',height=2.5)
```

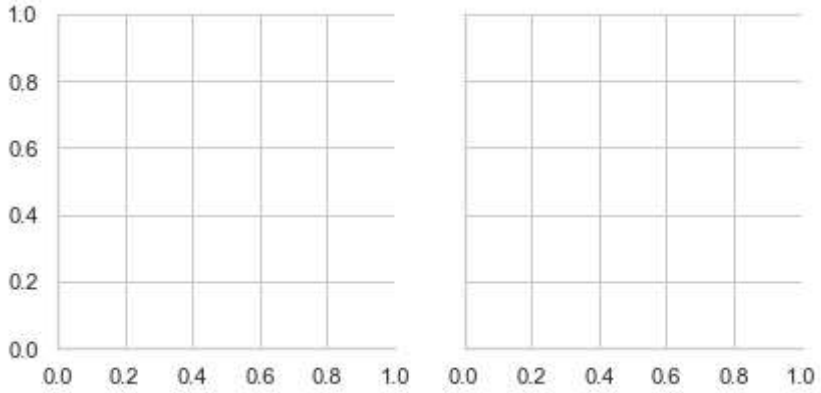
Out[69]:
```
<seaborn.axisgrid.PairGrid at 0x1db39b1a7f0>
```



使用FacetGrid時，我們會通過一個pandas DataFrame以及控制圖形網格的行、列和顏色的變量名稱來初始化一個對象。這些維度變量（控制行、列和顏色的變量）應該是分類變量或者離散變量，然後這些變量的不同水平組合起來就構成了整個圖形的每一個子圖（facet，在這裡可以理解為我們維度拆解的最小粒度）

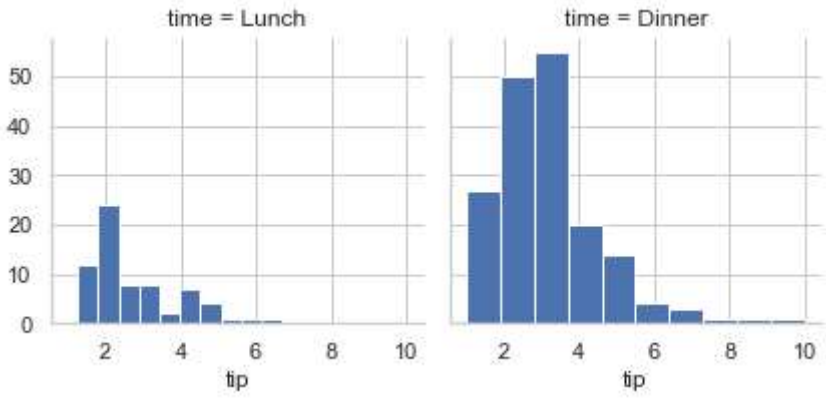
這一函數的目標是一步到位地提供一幅完整的成品圖，它在完成繪圖後還會對每個坐標軸添加注釋。

In [72]:
```
#沒給任意參數是空白
g = sns.FacetGrid(tips,col='time')
```



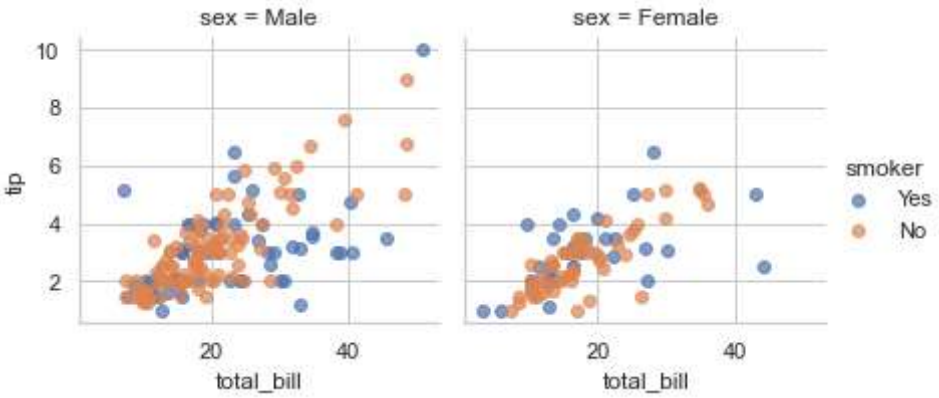
In [73]:
```
g.map(plt.hist,'tip')
```

Out[73]:
```
<seaborn.axisgrid.FacetGrid at 0x1db3e3b79a0>
```



In [75]:
```
g = sns.FacetGrid(tips,col='sex',hue='smoker')
g.map(plt.scatter,'total_bill','tip',alpha=.7)
g.add_legend()
```

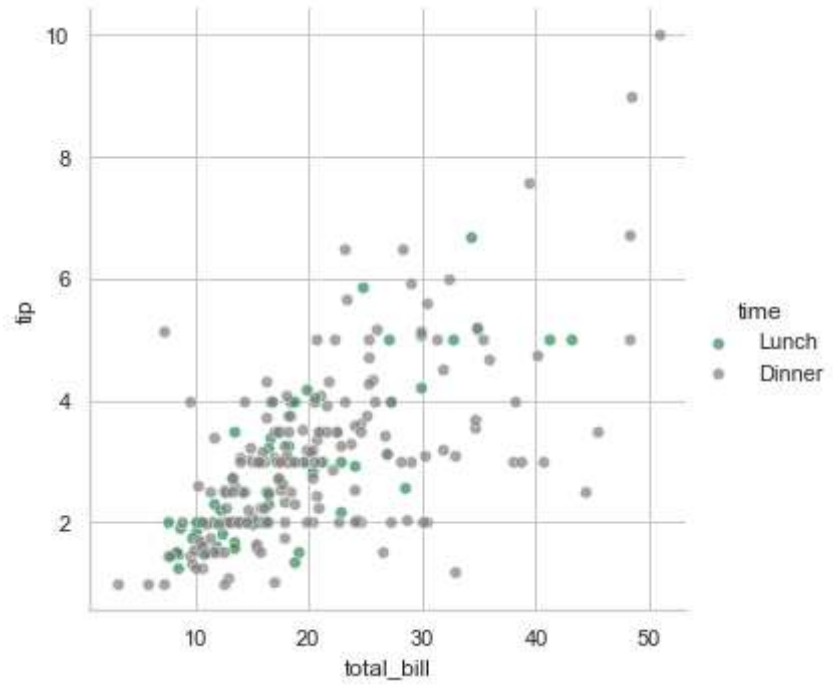
Out[75]:
```
<seaborn.axisgrid.FacetGrid at 0x1db3e31b0d0>
```



我們可以指定某個seaborn調色板，也可以通過字典將hue變量中的每個分類與其對應的matplotlib顏色傳遞給函數
（這樣就可以隨心所以使用大量的matplotlib支持的色彩）

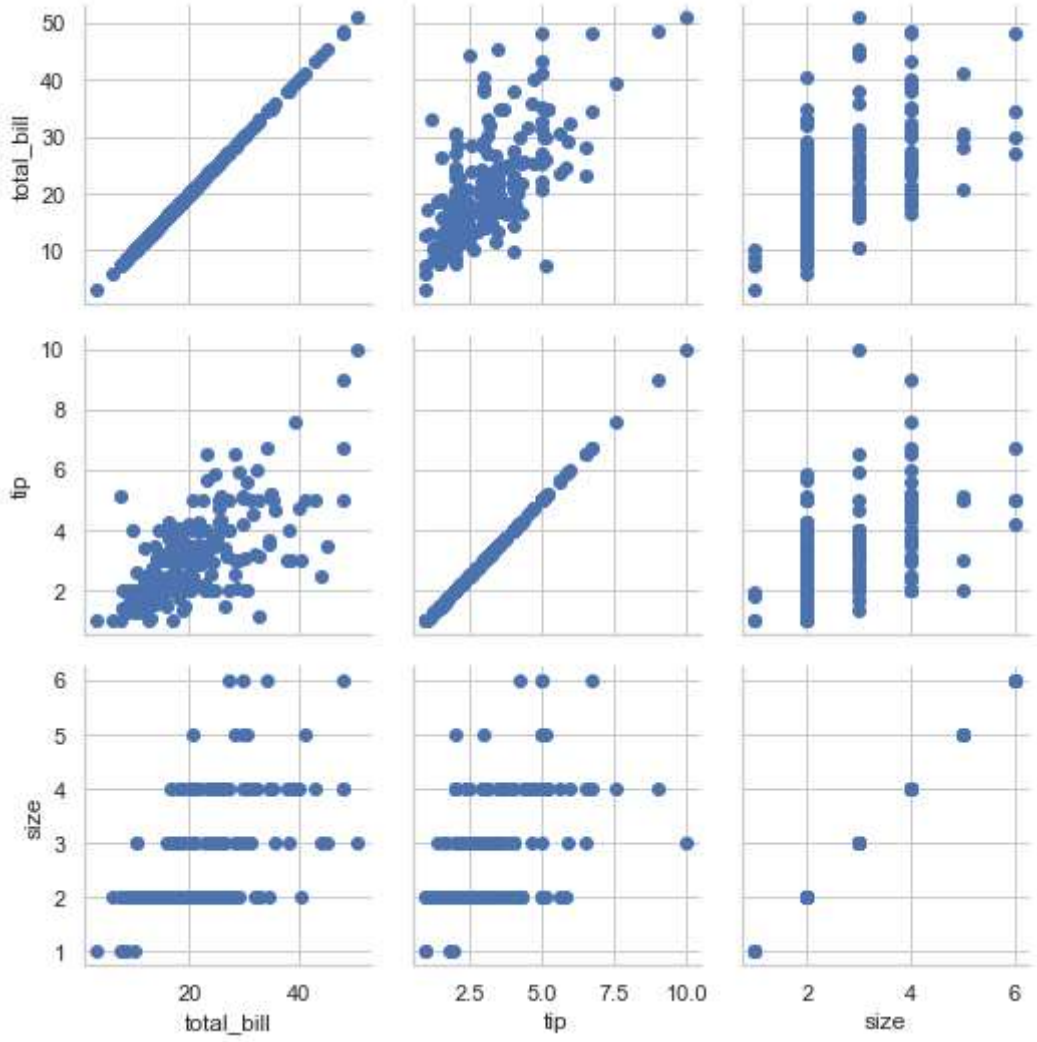
In [78]:
```
pal=dict(Lunch='seagreen',Dinner='gray')
g=sns.FacetGrid(tips,hue='time',palette=pal,height=5)
g.map(plt.scatter,'total_bill','tip',alpha=.7,linewidth=.5,edgecolor='whit
g.add_legend()
```

Out[78]: <seaborn.axisgrid.FacetGrid at 0x1db3e90ce80>



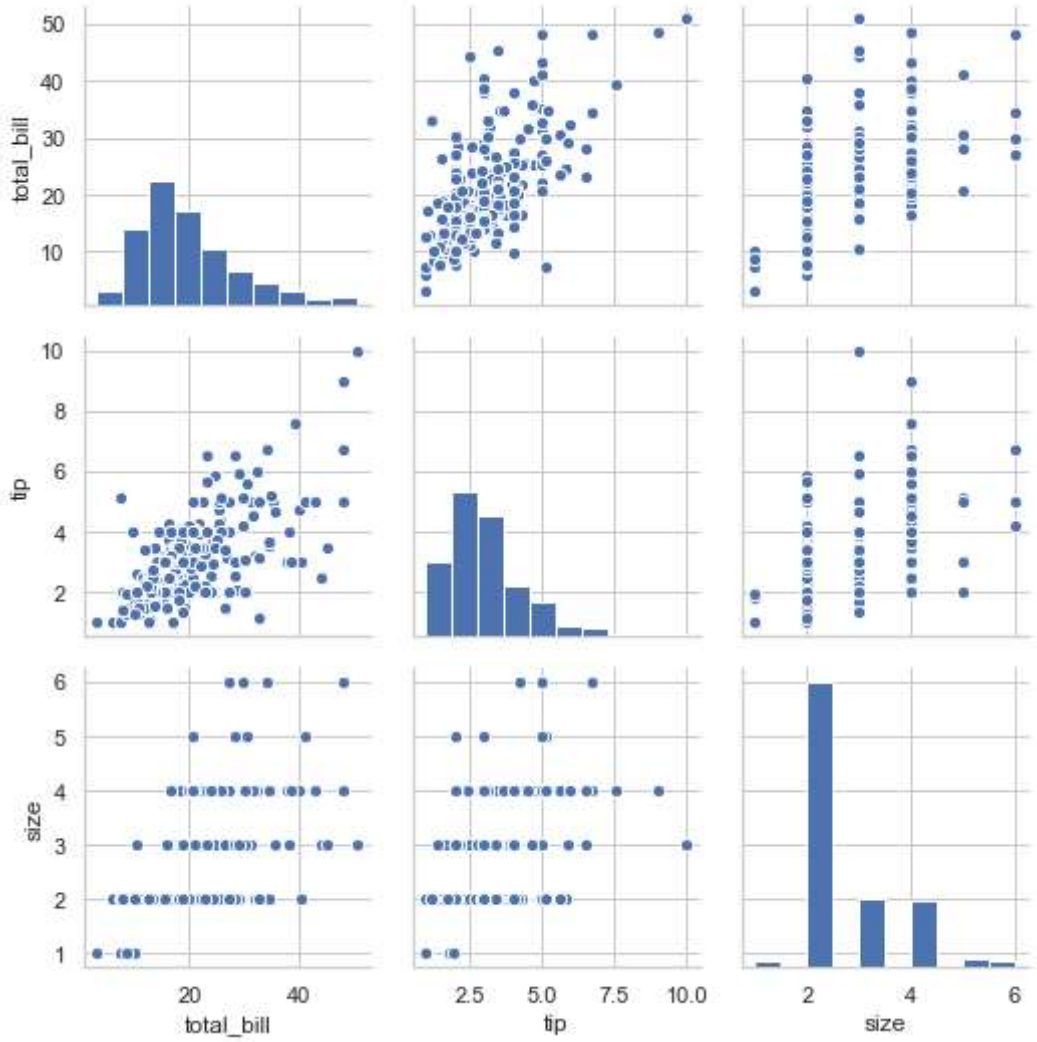
In [82]:
```
tips=sns.load_dataset('tips')
g=sns.PairGrid(tips)
g.map(plt.scatter)
```

Out[82]: <seaborn.axisgrid.PairGrid at 0x1db3fac8a60>



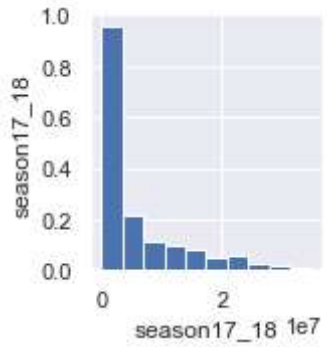
`map_diag()`函數是繪製對角線上的單變量子圖，`map_offdiag`是繪製對角線以外的兩個變量間的子圖。

In [87]:
```
g=sns.PairGrid(tips)
g=g.map_diag(plt.hist,edgecolor='w')
g=g.map_offdiag(plt.scatter,edgecolor='w',s=40)
```



[練習]試著讀取自己的資料，繪製出多維數據圖表

In [96]:
```
# Path of the file to read
fifa_filepath = 'data/NBA_season1718_salary.csv'

# Readd the file into a variable fifa_data
fifa_data = pd.read_csv(fifa_filepath, index_col="Unnamed: 0", parse_dates

g=sns.PairGrid(fifa_data)
g=g.map_diag(plt.hist,edgecolor='w')
g=g.map_offdiag(plt.scatter,edgecolor='w',s=40)
```



In [97]:
```
g=sns.PairGrid(fifa_data)
g.map(plt.scatter)
```

Out[97]: <seaborn.axisgrid.PairGrid at 0x1db43d5f4c0>